

### ***In the Claims***

The status of claims in the case is as follows:

1        1.    [Currently amended] A scalable system for providing a  
2        web processing tool, comprising:

3            a browser;

4            a first server cluster including a plurality of first  
5            clustered servers, each said first server running  
6            identical first mirror-image system and first  
7            application code for routing client requests among a  
8            plurality of enterprise applications;

9            a storage and execution unit, including a file server,  
10           a second server cluster, and an application server, for  
11           storing and executing an enterprise application  
12           comprising a plurality of web application code tables  
13           and data;

14           said file server including configuration file proxy  
15           statements for mapping user requests directed to said

16        web application to said second server cluster;

17        a ~~plurality~~ said second server cluster including a  
18        plurality of second ~~clustered~~ servers, each said second  
19        server running identical second ~~mirror image~~ system and  
20        identical selected components of said web application  
21        code, said second servers within said second ~~clustered~~  
22        ~~servers~~ server cluster periodically replicating with  
23        each other by moving identical data and code tables  
24        into identical data structures in each said second  
25        server so as to maintain data and code table  
26        consistency between them;

27        a relational database for storing data tables including  
28        a user profile specifying user roles;

29        said web application code tables including a first web  
30        application code table for authorizing access to other  
31        web application code tables based upon said user roles;

32        a database server for interfacing said relational  
33        database to said second server cluster and said  
34        application server;

35 a first network dispatcher for dynamically balancing  
36 client workload by redirecting ~~clients~~ client requests  
37 to one of said first ~~clustered~~ servers based on current  
38 workload of first servers within said ~~plurality of~~  
39 first ~~clustered servers~~ server cluster;

40 a second network dispatcher responsive said first  
41 clustered servers for dynamically balancing client  
42 workload by redirecting clients to one of said second  
43 clustered servers based on current workload of servers  
44 within said plurality of second clustered servers;

45 ~~an application~~ said application server asynchronously  
46 responsive to said second ~~clustered~~ servers for running  
47 agents to process application data requests and bridge  
48 said application data with respect to said database  
49 server and other back end servers.

1 2. [Currently amended] The scalable system of claim 1,  
2 said first ~~clustered servers~~ server being operable for  
3 presenting a graphical user interface to the said browser  
4 and for caching data on behalf of an end user.

1 3. [Currently amended] The scalable system of claim 1,

2     each said first ~~clustered servers~~ server being a domino.go  
3     ~~servers~~ server operable for presenting a graphical user  
4     interface to said browser and redirecting said client via  
5     said second network dispatcher to a second ~~cluster~~ server.

1     4.     [Original]   The scalable system of claim 1, said web  
2     processing tool being a web requisition catalog application.

1     5.     [Currently amended]   The scalable system of claim 1,  
2     said second ~~clustered~~ servers being operable for performing  
3     workflow, providing security, and serving as a document  
4     repository.

1     6.     [Currently amended]   The scalable system of claim 5,  
2     said second ~~clustered~~ servers being domino network servers.

1     7.     [Original]   The scalable system of claim 6, said  
2     document repository being requisitions stored in domino .nsf  
3     files.

1     8.     [Currently amended]   The scalable system of claim 2,  
2     further comprising an external objects dynamic file for  
3     storing external objects in one place for dynamic access by  
4     said first ~~clustered~~ servers, and for generating said gui.

1 9. [Original] The scalable system of claim 1, said  
2 database server being a relational database server.

1 10. [Original] The scalable system of claim 1, said other  
2 back end server comprising an enterprise resource planning  
3 system, including an accounting application having an  
4 accounts payable function.

1 11. [Currently amended] The scalable system of claim 3,  
2 further comprising a configuration file of proxy statements  
3 for mapping user requests to said second server cluster.

4 12. [Currently amended] A method for generating on-line  
5 procurement requisitions, comprising the steps of:

6 receiving a client request;

7 dynamically balancing client workload among a plurality  
8 of first logically defined servers in a first server  
9 cluster by directing said request to one of said first  
10 servers ~~a first server within a first cluster of~~  
11 ~~virtual servers~~ based on current server workload, each  
12 said first server running identical first system and

13        first application code for each server in said first  
14        cluster running first same application and system code,  
15        operating said first server to determine the mapping of  
16        distributing said client request and a required  
17        function the function required- among a plurality of  
18        enterprise applications including a database access  
19        function;

20        distributing said database access function an  
21        application server and in a plurality of code tables  
22        and data tables on a plurality of second logically  
23        defined servers in a second server cluster;

24        storing data tables including a user profile specifying  
25        user roles in a relational database;

26        responsive to a client request for said database access  
27        function, dynamically balancing client workload among  
28        said plurality of second servers in said second server  
29        cluster ~~directing said client request to a second~~  
30        ~~server within a second cluster of virtual servers based~~  
31        on current server workload;

32        operating one of said code tables to authorize access

33        to other of said code tables based on said user  
34        profile;

35        ~~, each server in said second cluster running second~~  
36        ~~same application and system code, servers within said~~  
37        ~~second clustered servers~~ periodically replicating said  
38        second servers with each other by moving identical data  
39        and code tables into identical data structures in each  
40        said second server so as to maintain data and code  
41        table consistency between them; and

42        operating said second server to direct said client  
43        request to an application server where all data is  
44        replicated and where bridges and agents execute with  
45        respect to data in said database.

1        13.    [Canceled]

1        14.    [Currently amended]    The method of ~~claim 13~~ claim 12,  
2        further comprising the steps of:

3                replicating application data to a back-end relational  
4                database server; and

5 replicating application data to a back-end enterprise  
6 resource planning system including an accounting  
7 application having an accounts payable function.

1 15. [Currently amended] A program storage device readable  
2 by a machine, tangibly embodying a program of instructions  
3 executable by a machine to perform method steps for  
4 processing a client request with respect to a database, said  
5 method steps comprising:

6 receiving a client request;

7 dynamically balancing server workload by directing said  
8 request to a first server within a first cluster of  
9 logically defined virtual servers based on current  
10 server usage, each server within said first cluster  
11 executing first ~~same code~~ application code for  
12 determining, ~~operating said first server to determine~~  
13 the mapping of said client request and the function  
14 required among a plurality of enterprise applications  
15 including a database access function;

16 responsive to a client request for said database access

17 function, dynamically balancing server workload by  
18 directing said client request to a second server within  
19 a second cluster of logically defined virtual servers  
20 based on current server usage, each server within said  
21 second cluster executing identical second ~~same code,~~  
22 application code including a plurality of web  
23 application workflow processes;

24 operating a first of said workflow processes for  
25 authorizing access to and execution of other of said  
26 workflow processes based on client user role;

27 periodically replicating said second servers within  
28 said second cluster of virtual servers periodically  
29 ~~replicating with each other by moving identical data~~  
30 and code tables into identical data structures in each  
31 said second server so as to maintain data and  
32 application code consistency between them; and

33 operating said second server to direct said client  
34 request to an application server where all data is  
35 replicated and where bridges and agents execute with  
36 respect to data in said database.

1 16. [Currently amended] A computer program product or  
2 ~~computer program element~~ for generating on-line procurement  
3 requisitions, said computer program product comprising:

4 a computer readable medium;

5 first program instructions for receiving a client  
6 request;

7 second program instructions for dynamically balancing  
8 server workload by directing said request to a first  
9 server within a first cluster of logically defined  
10 virtual servers based on current server usage, each  
11 server within said first cluster of servers executing  
12 identical first ~~same code~~ code;

13 third program instructions for operating said first  
14 server to determine the mapping of said client request  
15 and the function required;

16 fourth program instructions, responsive to a database  
17 access function, for dynamically balancing server  
18 workload by directing said client request to a second

19 server within a second cluster of logically defined  
20 second virtual servers based on current server usage,  
21 each said second server within said second cluster of  
22 logically defined second virtual servers executing  
23 identical second ~~same code~~ code tables, said second  
24 servers within said second cluster of second virtual  
25 servers periodically replicating with each other by  
26 moving identical data and code tables into identical  
27 data structures in each said second server so as to  
28 maintain data consistency between them; and

29 fifth program instructions for operating a code table  
30 in said second server to authorize said client request  
31 to access and execute selected other code tables based  
32 on client user role, and for operating said other code  
33 tables to direct said client request to an application  
34 server where all data is replicated and where bridges  
35 and agents execute with respect to data in said  
36 database; and wherein

37 said first, second, third, fourth, and fifth program  
38 instructions are recorded on said medium.

1 17. [Currently amended] The program storage device of

2 claim 15, said method further comprising the step of  
3 operating said first ~~clustered~~ servers for presenting a  
4 graphical user interface to the said browser and for caching  
5 data on behalf of an end user.

1 18. [Currently amended] The program storage device of  
2 claim 15, said method further comprising:

3 synchronizing all virtual servers within said first  
4 server cluster; and

5 synchronizing all virtual servers within said second  
6 server cluster.

1 19. [Previously presented] The program storage device of  
2 claim 18, said method further comprising:

3 replicating application data to a back-end relational  
4 database server; and

5 replicating application data to a back-end enterprise  
6 resource planning system including an accounting  
7 application having an accounts payable function.

1     20. [Previously presented] The program storage device of  
2     claim 15, said method further comprising storing external  
3     objects in an external objects dynamic file for dynamic  
4     access by said first cluster of servers, and for generating  
5     said gui.